Research of 3D part reconstruction from Orthographic Projections Based on Solid Features

HAO Yong-tao, Xu Wenyi

Abstract A new method based on solid features for automatic reconstruction is proposed in this paper. The main features of the algorithm are to improve the speed of the reconstruction process. A 3D object is composed of one, or more than one part. The idea of the algorithm is to extract closed loops from orthographic views and to match loops from two views in order to build features, which are then combined by Boolean operations so as to form the 3D object finally. The algorithm does not match all loops of views, while redundant loops are deleted when feature is found. The example from a working implementation is given to demonstrate the efficiency and capability of the proposed algorithm.

Keywords 3D object, Orthographic projections, Feature, Algorithm

1 INTRODUCTION

Over the past three decades, reconstruction of 3-D solid models from 2-D engineering drawings has been applied and researched widely, and many different methods are proposed. They are “boundary representation (B-Rep)”, known as the bottom-up approach and “constructive solid geometry (CSG)” approach. The major steps of B-Rep method are illustrated as follows:(1) generate 3-D vertices from 2-D points; (2) generate 3-D edges from 3-D vertices; (3) generate 3-D faces from 3-D edges; (4) assemble faces into an object. From this algorithm, false elements not belonging to the object can be generated as well. On the other hand, the CSG approach assumes that each 3-D object can be built from certain primitive in binary tree. It selects a 2-D loop as base and uses translational sweep operations to construct a 3-D primitive; the constructed primitives are then combined to form the final object by using Boolean operators: union, intersection, and difference. The algorithm appears to handle only single solution case. Along with the increment of the number of loop in orthographic views, the complexity of reconstruction will be increased more quickly, and how to speed up the efficiency of current reconstruction algorithm is the main content of this paper. The algorithm proposed in this paper does not match all loops of views, and redundant loops are deleted when feature is found. Therefore, the efficiency of reconstruction will be accelerated clearly.

2 DEFINITION AND TERM

2.1 Definition

Symbols of views: top (T), front (F), and right (R); f, t, r: vertex of view; v: vector of view; L: loop of view; ↑: extrusion parameters separator; L(x ↑ (x₁ → x₂)): extrusion from x₁ to x₂ in the direction of x axis; Fe: feature;

2.2 Term

Group view: It consists of two different views among Front view, Top view, Right view, such as (F, T). The former called the Main View of the group, and the latter called the Subordinate View of the group.

Common coordinate axis: The coordinate axis that is common between group views. In Figure1, for example, the z-axis is the common coordinate axis between (a) and (b).

Extreme points: Points on a circle (ellipse, arc) with the maximum or minimum coordinate values along the common coordinate axis called as the extreme points of it. If the point p has the maximum coordinate value along the common coordinate axis, p is called as the maximum point. If the point p has the minimum coordinate value along the common coordinate axis, p is called as the minimum point. In Figure2, p₁, p₄ are the maximum points of circle, and p₂, p₃ are the maximum points of circle along the common
coordinate, respectively.

**Minimum vector:** The vector that can not be divided. In Figure 1(b), \(v(f_2, f_3), v(f_3, f_4)\) are minimum vectors.

**Basic vector:** The vector that consists of one or more than one minimum vector, such as \(v(f_2, f_4)\) in Figure 1(b).

**Point set of loop:** A set that consists of endpoints of vectors in a loop.

\[
\begin{align*}
V_2 &= v(f_2, f_3), v(f_4, f_5), v(f_7, f_6), v(f_3, f_6), v(f_8, f_6), \text{...}
V_3 &= v(f_6, f_4), \text{...}
V_{11} &= v(f_1, f_6), v(f_3, f_5), v(f_1, f_5), v(f_1, f_4), v(f_1, f_3), \text{...}
V_{12} &= v(f_1, f_3), v_6(f_2, f_4), v_7(f_2, f_5), v_8(f_4, f_3), v_9(f_2, f_6), v_10(f_2, f_5), v_11(f_2, f_4), \text{...}
\end{align*}
\]

**Point set of loop:** A set that consists of endpoints of vectors in a loop.

3 THE RECONSTRUCTION ALGORITHM

3.1 Foundation of reconstruction

This algorithm mainly process features of the main direction. A feature can be constructed by a 2-D loop, extruded direction and extruded distance. The constructed features are then combined to form the final object by using Boolean operators; union, intersection, and difference.

3.2 Pre-processing of source data

Source data of the algorithm must satisfy certain rule that it must be consisted of points or minimum vectors of orthographic views. It can be illustrated as follows:

1) Convert circle, ellipse, and arc into vector.
   As circle is symmetric graphic, it can be converted into a rectangle, using four extreme points of it. Regarding ellipse, the algorithm requires that any one of its axis should parallel to the corresponding coordinates. The method of processing data is the same as circle. For arc, assume that the arc and the remaining vector connecting arc composed of closed region \(S_1\). The vector that connected the arc’s two endpoints and the remaining vectors connecting the arc composed of closed regional \(S_2\). Projection ranges of region \(S_1\) on the coordinate axis are called as \(I_1\), and \(I_2\). Projection ranges of region \(S_1\) on the corresponding coordinate’s axis projection are called as \(I_3\), and \(I_4\).

And the algorithm requires that one of the formulas \((I_1 \cup I_3 = I_3, I_2 \cup I_4 = I_4)\) is true. We can get a closed region which consists of vectors by connecting the arc’s two endpoints.

For the drawing represented by Fig.2 (a1), (b1), (c1), three drawing consisted of vectors can be generated, respectively, as shown in Fig.2 (a2), (b2) and (c2).

(2) Minimization

It means to break down the vector into smallest units according to the intersection between vectors in orthographic views.

![Figure 1 Orthographic view of form](image)

![Figure 2 example 1](image)
In Fig. 3, it just need eight vertexes ($f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8$) and six vectors ($v(f_1, f_3), v(f_3, f_5), v(f_5, f_7), v(f_7, f_1), v(f_2, f_6), v(f_4, f_8)$) to describe the graphics clearly. The algorithm of transformation is illustrated as follows:

**Algorithm 1:** minimization.

Output result: all points and vectors of views.

//source data: points set $P$, vectors set $V$;
//result set: $R$;
while ($V !=$ null and $V.Count > 1$)
{
    $V.Select(v_1)$; //choose $v_1$ in $V$ randomly
    I.Clear();
    while ($v$ in $V$)
    {
        if ($isIntersect(v_1,v) == true$ && $p != v_1$ vertex) // $p$ is the intersect point
        {
            if ($P.NotExists(p)$)
            {
                $P.Add(p);$
            }
            I.Add($p$); //mark this point
        }
    }
    If ($I !=$ null)
    {
        $R.Add(v_1.Split(I));$
    }
    else
    {
        $R.Remove(v_1);$ 
    }
$V.Remove(v_1);$ 
}

For Fig. 3, after the execution of algorithm 1 we can get all nine points and 12 minimum vectors.

![Figure 3 example 2](image)

### 3.3 Searching for the loops in views

**Algorithm 2** Searching for loops which consist of minimum vector.

An undirected graph can be generated by vertexes, vectors and relationships between vectors. Using the method of DFS (Depth First Search) and RST (Recursive Search Technique), loops in views can be further constructed from graphs. The methods are as follows:

1. Store vectors and relationships in views into adjacency table. For example, in Fig. 1 (b), $v_{f_1}, v_{f_4}$ are described as follows: $v_{f_1} \rightarrow v_{f_2} \rightarrow v_{f_3}, v_{f_4} \rightarrow v_{f_5} \rightarrow v_{f_6}$, respectively.
2. Starting from a node randomly, DFS is used to search for loops. During the process, the nodes on the path of searching will be pushed into stack I.
3. If the number of option nodes in the adjacent table is larger than one, single one will be selected randomly as a successor. Other nodes and the current node will also be pushed into stack II in pairs.
4. When the successor searched is equal to the node which belongs to the nodes stored in stack I, the search for a loop is over. All nodes of this circuit will be stored in the database of loops. Then, the last recorded pair is popped from the stack II, and the pair becomes the new starting node.
5. While converting information of views into undirected graph, and if there are three or more vectors intersect at one vertex, there is an invalid loop in the undirected graph. In the process of traversing, traverse should be interrupted if the subsequent node and nodes in stack I compose of invalid loop. And node should be trackback to the top node of stack II to start a new traverse.
6. When there are no nodes stored in the stack II, the search process is over, and the database of loops has collected all the circuits searched.

**Algorithm 3** Constructing loops based on basic vectors by merging minimum vectors in the loops constructed through algorithm 2.

Traversing minimum vectors of each loop, it can be merged into basic vector while the adjacency vectors ($v_1, v_2$) have the same slope. A new vector (basic vector) will be formed by connecting with $v_1$ and $v_2$.

After the execution of algorithm 3 we can get all loops constructed by basic vectors.

### 3.4 Recognition of main direction features

Choose the view on the main direction as the main view, and choose anyone of the remaining views as subordinate view; they compose of a group of view. Find outline in the main view, and find extruded
direction and extruded distance in the subordinate views. The specific steps are illustrated as follows:

1. In accordance with loop $L$ in the main view, sought the projective coordinates set of the loop point set in common coordinate axis.
2. According to these projective coordinates, we can get mapping point set of each loop in the subordinate views, respectively.
3. According to the mapping point set, we will find that whether a mapping range exists in non-common coordinate axis of subordinate views or not.
4. If there is a mapping range exists, a feature of main direction is found. The loop $L$ is the outline, the mapping range is the extruded direction, and the length of mapping range is the extruded distance. Otherwise, the above steps will be repeated until traverse finished.

3.5 Solid reconstruction

During the process of solid reconstruction, the Boolean relationship can be found in accordance with the relationship among the loops. The rules are as follows:

**Rule 1** Rules of Boolean operations

- if $L_1 \cap L_2 = L_1$ or $L_1 \cap L_2 = L_2$, then Result $= F_{e_2} - F_{e_1}$ or Result $= F_{e_1} - F_{e_2}$;
- if $L_1 \cap L_2 = \phi$, then Result $= F_{e_2} \cup F_{e_1}$;
- otherwise, Result $= F_{e_2} \cap F_{e_1}$.

In addition, in order to reduce unnecessary searching and to make the algorithm more efficiency, all loops which can be found by the constructed features will be removed from the views immediately during the process. The specific algorithms are illustrated as follows:

**Algorithm 4** According to the constructed solid, we can get the corresponding vertex and vectors of it. After the extend operation of the loop, the vertex can be formed of two parts: the vertex of the loop itself and the vertex of the symmetric loop form by extend the loop to the terminal plane. The vector can be formed of three parts: the composing vector of the loop itself; the composing vector of the symmetric loop form by extends the loop to the terminal plane and the vector formed in process of extends every vertex of the loop.

**Algorithm 5** Removing loops.

According to algorithm 4, we can get the corresponding vertex and vectors of a new feature which found by step 3.4. Then, we can find that in the views whether there is loop that all of its composed vectors can be expressed by the new feature. Remove it if it exists.

**Algorithm 6**

According to the algorithm 4, all the vertex and vectors of the reconstructed solid should be compared to the original orthographic views. If any point or any vector in the orthographic views can be described by reconstructed features, the reconstruction is finished; otherwise, searching for features will be continued.

3.6 Other rules

**Rule 2** If there are many loops in the main view with the same projection coordinate set, the loop with the biggest outline will be processed firstly.

**Rule 3** Regarding a loop in the main view, if there are many mapping interval in the subordinate view, the mapping interval which has maximum length will be processed firstly.

4 FLOW OF RECONSTRUCTION

During the process of searching for features of main direction, the loop formed by circle, ellipse, and arc must be done firstly. Usually, the feature formed by these objects is very clearly. For example, a circle in the main view and a rectangle in another view indicate that it is a cylinder feature.

Secondly, we can choose the main direction according to the number of bias vector, which is to choose the view with most bias as the main direction. To find features in the main direction according to the above algorithm, if the seeking on that direction is over but reconstruction is not finished yet, we should switch the main direction by choosing the view with second most number of biases, and continue until reconstruction operation is over.

5 EXAMPLE

Fig.1 will be taken as an example to illustrate the process of reconstruction. Firstly, points and minimum vectors set got by standardized data are as Fig.1 (e) indicated.

After executing of algorithm 2 and algorithm 3, we
can get loops consist of minimum vectors, new vectors which shown in Fig.1 (f), and loops consist of basic vectors. The loops are as follows:

$L_{F1}: \ V_{1,5} ' V_{2,2} ' V_{1,5} v_{1,2} ' V_{3,5} ' V_{2,5} ' V_{3,7} ' V_{8}$
$L_{F2}: \ V_{1,5} ' V_{2,7} ' V_{3,5} ' V_{3,7} ' V_{3,14} ' V_{3,15} ' V_{3,12} ' V_{2,12} ' V_{1,5} ' V_{2,5} ' V_{2,7} ' V_{2,9} ' V_{3,7} ' V_{8}$
$L_{F3}: \ V_{1,5} ' V_{3,15} ' V_{3,17} ' V_{7,10} ' V_{7,8}$
$L_{F4}: \ V_{1,5} ' V_{1,10} ' V_{1,7} ' V_{1,9} ' V_{7,10}$
$L_{F5}: \ V_{1,5} ' V_{1,10} ' V_{1,7} ' V_{1,9} ' V_{7,10}$
$L_{F6}: \ V_{1,5} ' V_{1,10} ' V_{1,7} ' V_{1,9} ' V_{7,10}$
$L_{T1}: \ V_{1,15} ' V_{3,15} ' V_{7,16} ' V_{7,8}$
$L_{T2}: \ V_{1,15} ' V_{3,10} ' V_{7,11} ' V_{7,16} ' V_{7,8}$
$L_{T3}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,8}$
$L_{T4}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{T5}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{T6}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{T7}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{T8}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{T9}: \ V_{1,15} ' V_{7,11} ' V_{7,7} ' V_{7,9}$
$L_{L1}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L2}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L3}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L4}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L5}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L6}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L7}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L8}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L9}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L10}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L11}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L12}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L13}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L14}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$
$L_{L15}: \ V_{9,1} ' V_{8,2} ' V_{1,23} ' V_{2,26}$

Secondly, We find $L_{F1}$ in the top view. So the main direction is top view. Point set and projection coordinate set of loop $L_{T1}$ are shown in Table1.

Mapping point set is shown in Table2.

<table>
<thead>
<tr>
<th>Loop</th>
<th>Points of Loop</th>
<th>Projection of x axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_{T7}$</td>
<td>$t_{10}, t_{11}, t_{7}$</td>
<td>$[5, 6]$</td>
</tr>
</tbody>
</table>

Table2 corresponding points of $F$

<table>
<thead>
<tr>
<th>$x$</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

We can find a mapping interval $[3.5, 5]$, and outline ($L_{F3}$), starting plane ($z = 3.5$), terminal plane ($z = 5$), and extruded length ($1.5$) could be found together. As a result, the feature $F_{T7} = L_{F3}$ ($z = 3.5, 5$) will be constructed and shown in Fig.5(b).

The feature $F_{T7}$ has already been reconstructed as the first feature of entity. The vertex set and the vector set covered by the feature will be obtained by executing algorithm 4. However, we find that point $f_1(0, 0)$ in $F$ view is not involved in the points set of the reconstructed feature by executing algorithm 6. After removing loops ($L_{F6}, L_{F7}, L_{F10})$ by executing algorithm 5, the process of searching features must be continued.

Then, $F$ view is regarded as the main direction because it contains a bias vector and other views do not have. (Front, Top) will be selected as a group view randomly. Point set and projection coordinate set of loops in front view are shown in Table3. Mapping point set of top view is shown in Table4.

Table3 Loop of $F$

<table>
<thead>
<tr>
<th>Loop</th>
<th>$L_{F1}$</th>
<th>$L_{F2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points of loop</td>
<td>$f_1, f_2, f_3, f_6, f_8$</td>
<td>$f_1, f_2, f_3, f_6, f_8$</td>
</tr>
<tr>
<td>projection</td>
<td>${0, 1, 3, 8}$</td>
<td>${0, 1, 3, 5, 6, 8}$</td>
</tr>
</tbody>
</table>

Table4 corresponding points of $T$

<table>
<thead>
<tr>
<th>$x$</th>
<th>0</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_-$</td>
<td>$t_{1}(0, 0)$</td>
<td>$t_{3}(1, 0)$</td>
<td>$t_{3}(3, 0)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$x$</th>
<th>5</th>
<th>6</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_-$</td>
<td>$t_{1}(0, 0)$</td>
<td>$t_{3}(1, 0)$</td>
<td>$t_{3}(3, 0)$</td>
</tr>
</tbody>
</table>

Using the same method, we will find that loops ($L_{F1}, L_{F2}, L_{F3}$) in Table 3 can not find a mapping interval in Table 4, so they are removed directly.

Regarding loop $L_{F4}$, a mapping interval $[0, 6]$ can be found in Table 4, and outline ($L_{F4}$), starting plane ($y = 0$), terminal plane ($y = 0$), and extruded length ($6$) could be found. As a result, the feature $F_{T4} = L_{F4}$ ($y = 0$)
↑ (0, 6)) will be constructed and shown in Fig.5 (a).

According to rule 1, the relationship between loop $L_{F_4}$ covered by $Fe_T$ and loop $L_{F_5}$ covered by $Fe_F$ is $L_{F_4} \cap L_{F_6} = L_{F_6}$, so we can get the formula $Result = Fe_F - Fe_T$.

The vertex set and the vector set covered by the updated feature will be obtained by executing algorithm 4. However, we find that point $f_9$ (3, 5) in $F$ view is not involved in the points set of the reconstructed feature by executing algorithm 6. After removing loops $(L_{F_4}, L_{R_1}, L_{R_5}, L_{R_6}, L_{R_8}, L_{R_{10}}, L_{T_7}, L_{T_8})$ according to feature $Fe_F$ by executing algorithm 5, the process of searching features must be continued.

Regarding loop $L_{F_5}$, a mapping interval $[0, 2]$ can be found in Table 4, and outline $(L_{F_5})$, starting plane $(y=0)$, terminal plane $(y=2)$, and extruded length (2) could be found. As a result, the feature $Fe_{F_3} = L_{F_5}(y \uparrow (0, 2))$ will be constructed and shown in Fig.5 (c).

According to rule 1, the relationship between loop $L_{F_4}$ covered by $Fe_T$ and loop $L_{F_5}$ covered by $Fe_F$ is $L_{F_4} \cap L_{F_5} = L_{F_5}$, so we can get the formula $Result = Fe_F - Fe_{F_3}$. And it is shown in Fig.5 (d).

The vertex set and the vector set covered by the updated feature will be obtained by executing algorithm 4. Executing the algorithm 4 again, we will find that reconstruction is over. The final solid of reconstruction is indicated as Fig.5 (e).

### 6 CONCLUSION

This paper introduces a solid reconstruction method based on features of main direction. The method is to extract closed loops from orthographic views and to match loops from two views in order to build features. These features are then combined by Boolean operations so as to form the 3D object finally. The method does not match all loops of views, and redundant loops are deleted when feature is found. Several tests for sample models have been conducted to verify the effectiveness of the proposed method.

However, the geometric domain is limited to extruded features. Reconstruction of solids of revolution needs further work.

### REFERENCES


